

# Ansible Automation on IBM i

## Benoit MAROLLEAU – Cloud/AI Architect

IBM Systems Center Europe, Montpellier, France  
[benoit.marolleau@fr.ibm.com](mailto:benoit.marolleau@fr.ibm.com)



Replays & Presentations  
<https://ibm.biz/bma-wiki>

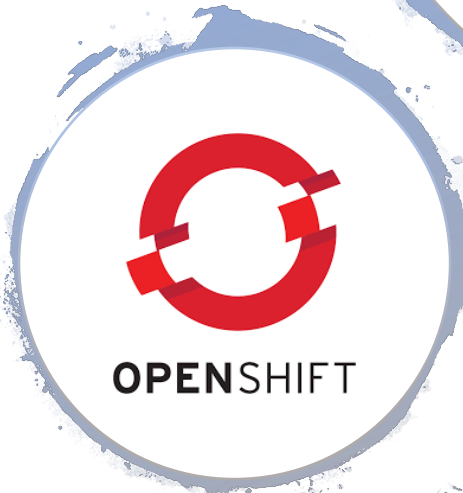


OPENSIFT



# Agenda

- Infrastructure as Code Strategy
- Ansible 101
- Ansible & IBM i
- Demonstration

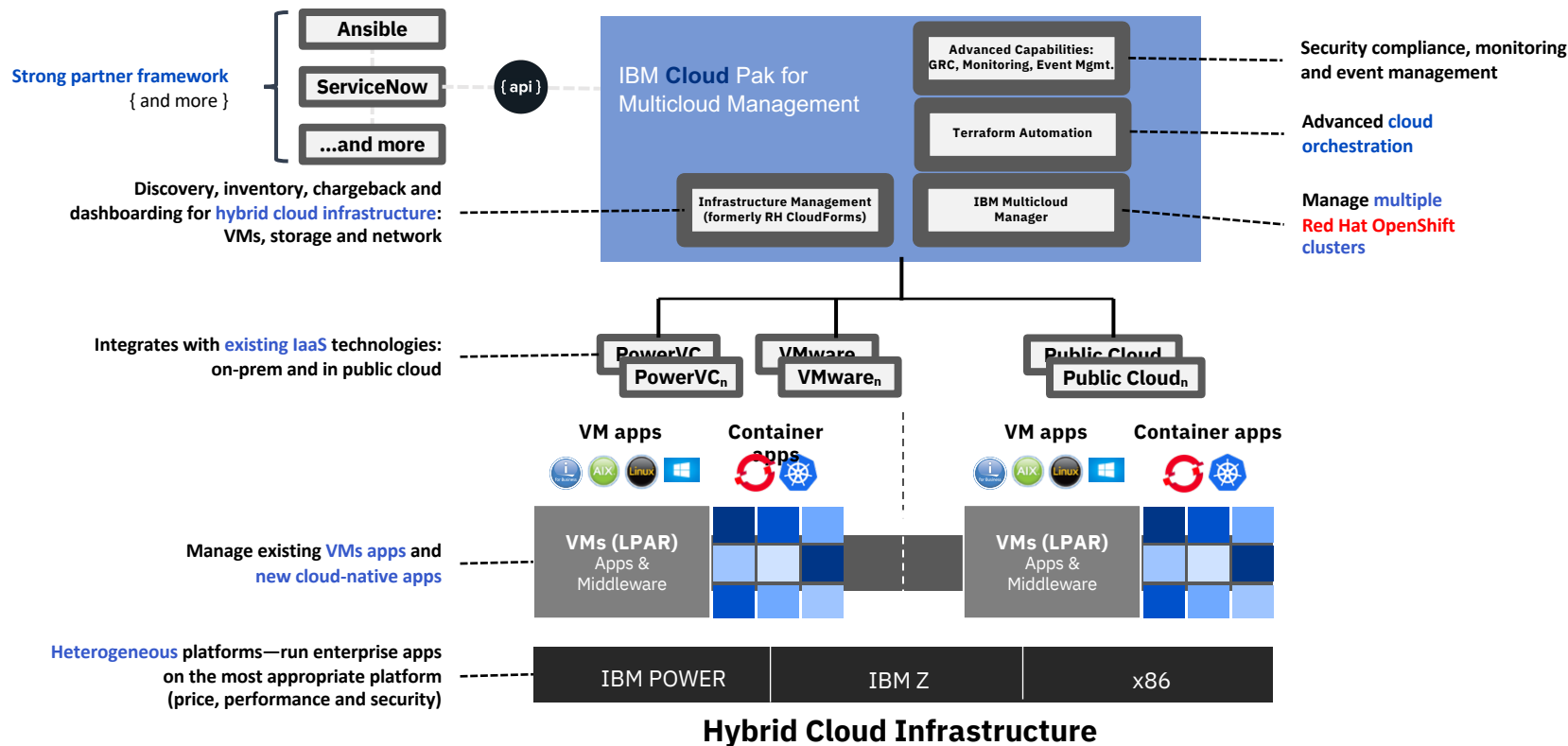


# Infrastructure as Code ?

---

<https://www.ibm.com/cloud/blog/chef-ansible-puppet-terraform>

# Hybrid Cloud Architecture with IBM Cloud Pak for Multicloud Management





# IBM Multicloud Manager

## Visibility

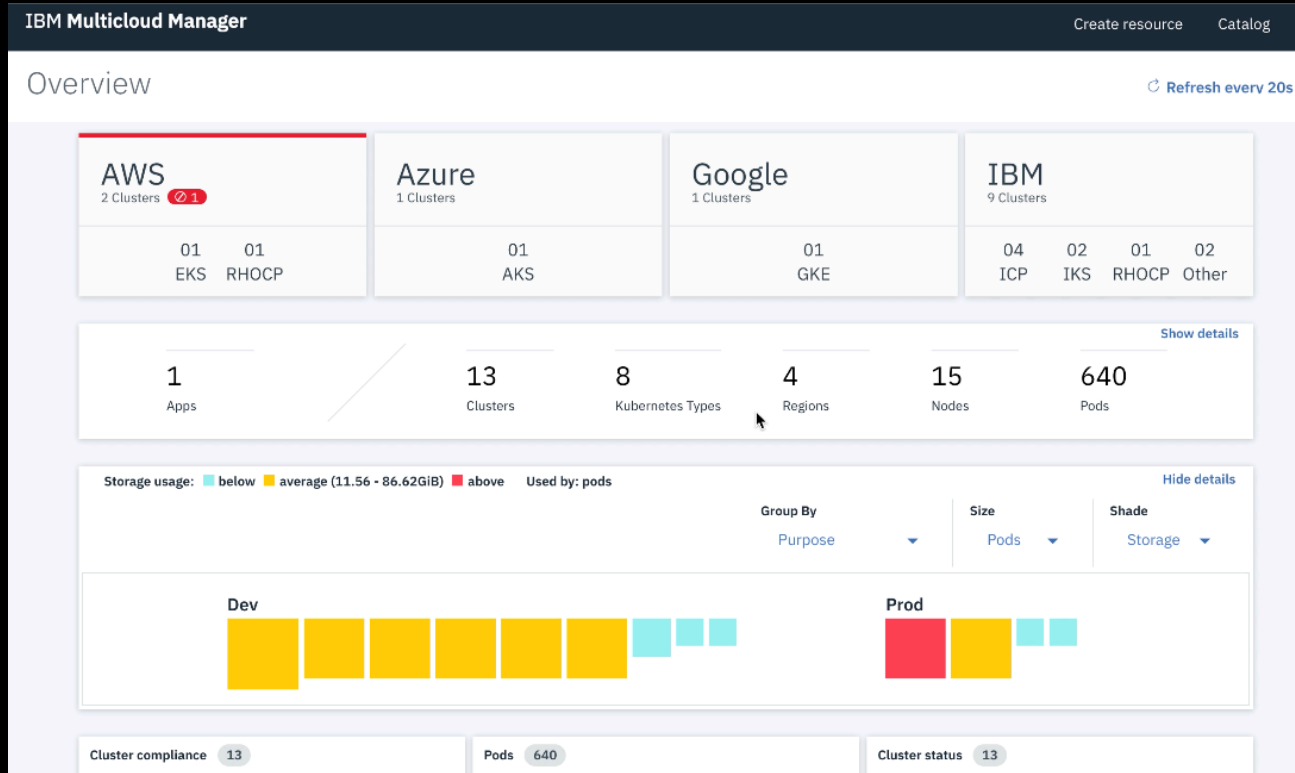
- Development teams can see all the deployed pods
- IT Operations can see clusters and nodes

## Security and Governance

- Consistent configuration and security policies across cluster

## Automation

- Automatically provision, configure and deliver additional Kubernetes clusters in any cloud environment supported by Cloud Automation Manager



# Cloud Pak for MCM

## Multiple Cloud Connectors : VMWare, PowerVC, Public Clouds

IBM Cloud Automation Manager

Manage

Cloud Connections

Content Runtimes

Email Configuration

Shared Parameters

Ansible Automation

Terraform Versions

Cloud Connections

Cloud Connections allow you to create connections to each of your target cloud providers that are used when deploying instances.

Cloud Connections (5)

Search Connections

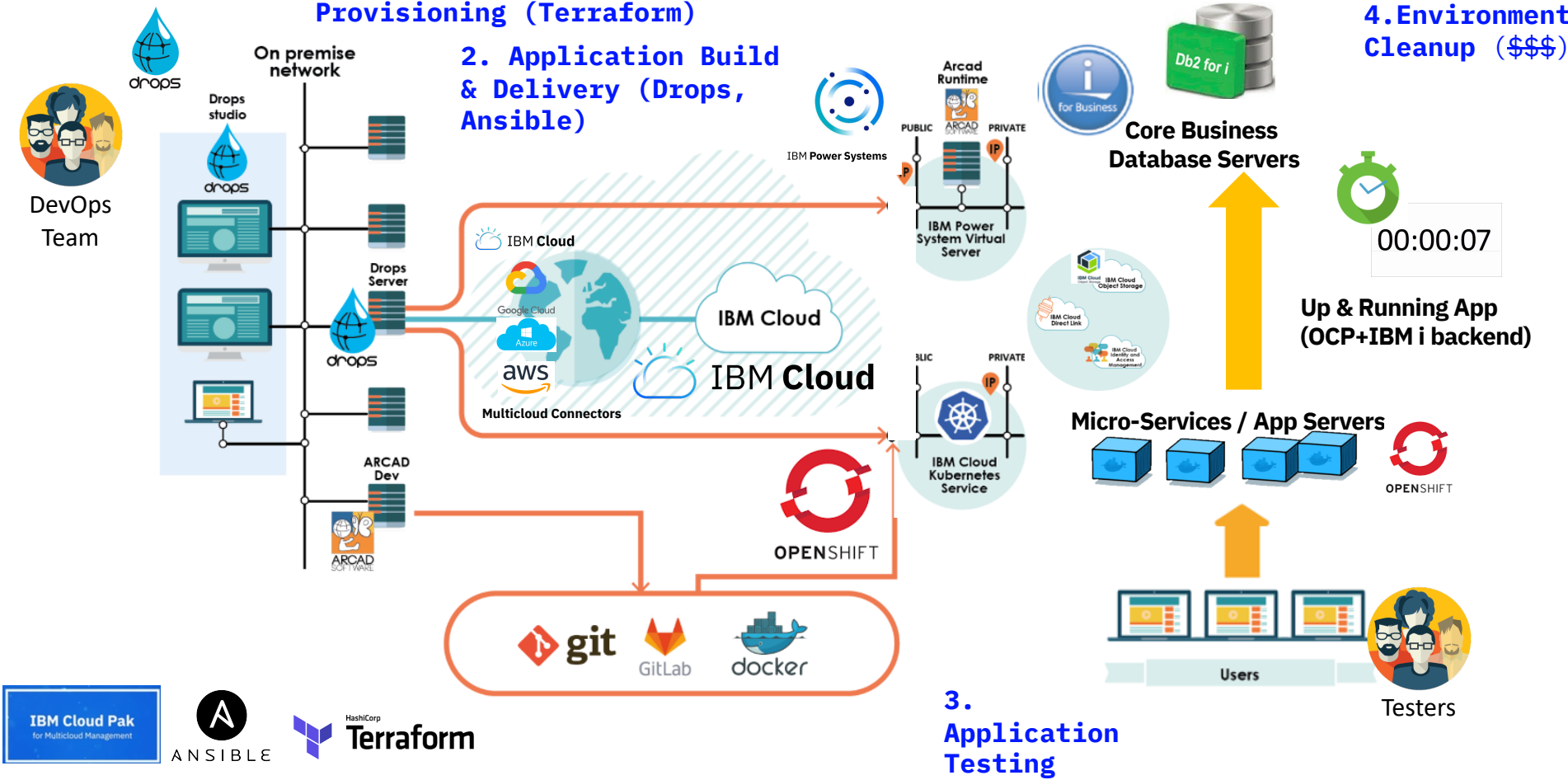
Name ▲	Cloud Provider	Namespace	Status
IBMCloud	IBM	Globally Accessible	Valid
IBMPowerVC	OpenStack	Globally Accessible	Valid
PowerVC	OpenStack	Globally Accessible	Valid
PowerVC-Prod	OpenStack	Globally Accessible	Valid
vcentersandbox	VMware vSphere	Globally Accessible	Valid

# MultiCloud CI/CD with Arcad Drops on IBM Cloud / Power Virtual Server

## 1. QA/Dev/Test Infrastructure Provisioning (Terraform)

## 2. Application Build & Delivery (Drops, Ansible)

## 4. Environment Cleanup (\$\$\$)



# Ansible

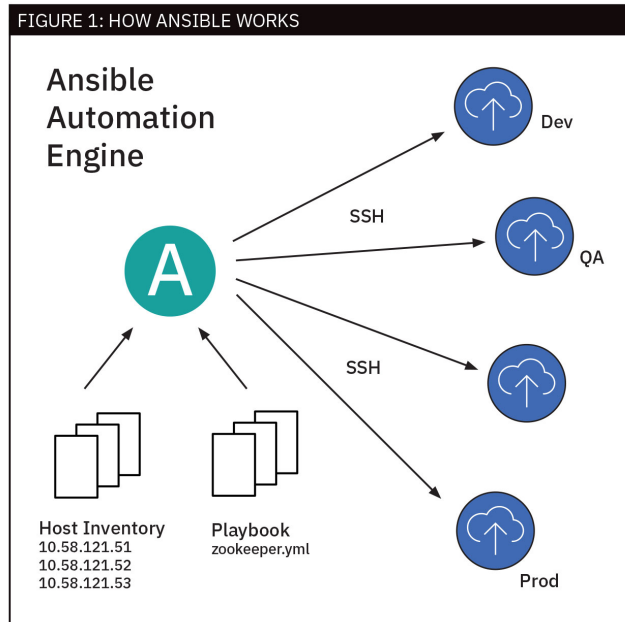
---

# Introduction to Ansible



Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy.

- Free open source application
- **Agent-less** – No need for agent installation and management
- Python/YAML based
- Highly flexible and configuration management of systems.
- Configuration roll-back in case of error



# Introduction to Ansible

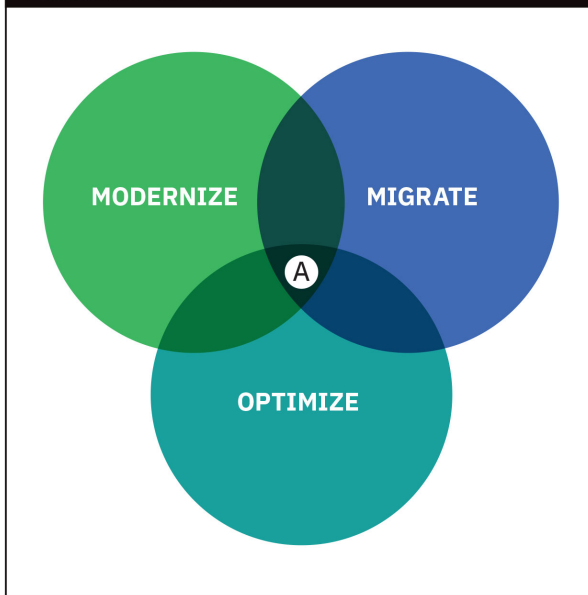


Common use cases are:

- ✓ Automation of repeatable operations tasks like **system administration**, code build, deployment and QA tests in your DevOps cloud infrastructure and cloud deployment pipeline
- ✓ **Cloud infrastructure and application provisioning**, complementing other infrastructure as code technologies like Terraform
- ✓ **Configuration and security compliance** reinforcement by checking and fixing systems settings versus policies in place

IBM i Operations like any other platforms...  
Efficient for new hires onboarding !!

FIGURE 2: USE ANSIBLE TO MODERNIZE, MIGRATE AND OPTIMIZE YOUR CLOUD DEPLOYMENTS



# Introduction to Ansible



**Control node** – any machine with Ansible installed and is used to run playbooks



**Managed node (a.k.a. endpoints)** – endpoint devices (e.g., AIX, IBM i, Linux, Windows, etc.) that are managed with Ansible



**Inventory** – a list of managed nodes so that Ansible understands the overall IT landscape



**Modules** – units of code that Ansible executes; [hundreds of modules out-of-box](#); thousands of community modules available



**Tasks** – units of action in Ansible  
(invoke a set of modules to do something useful)



**Playbooks** – ordered list of tasks and written in YAML



# Introduction to Ansible

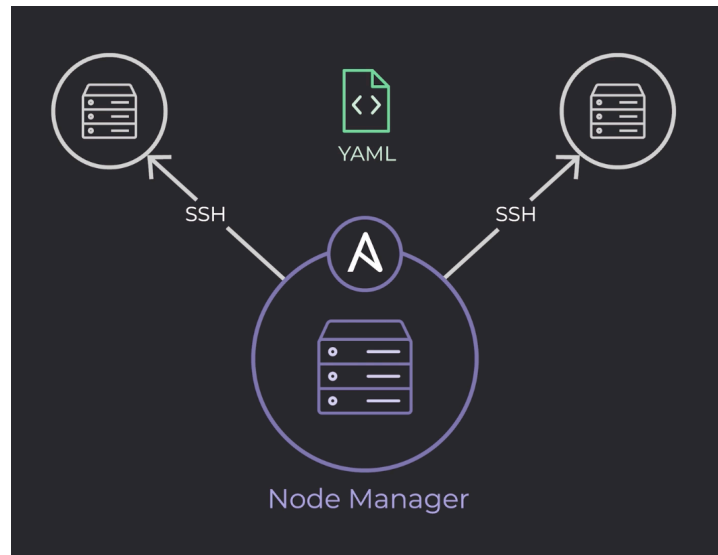
## Idempotency , Parallelism

Ansible is **idempotent**, it means that an operation has the same effect, whether applied one or more times. If the state requested in the action is in accordance with what is requested, then Ansible does nothing.

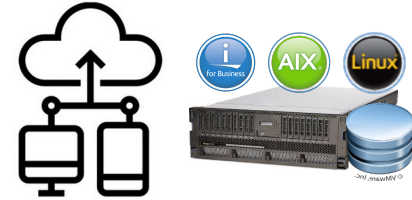
Idempotency is possible because Ansible manages 3 execution states for a task : **skipping**, **ok** or **changed**.

Ansible launches the tasks in push mode (in parallel) on all the servers at the same time. On the other hand, the tasks are executed sequentially one after the other.

The first task will be sent and executed on all servers before proceeding to the next task. This means that the node manager waits for a response from all the nodes and waits until they have all completed the execution of the current task before proceeding to the next one.



# Introduction to Ansible



Cloud  
Infrastructure



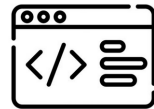
Application



Security



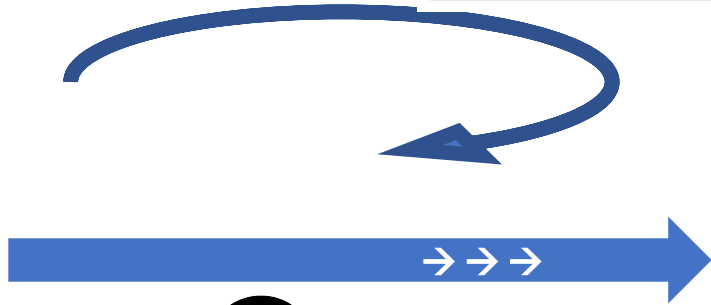
Templates



Scripts

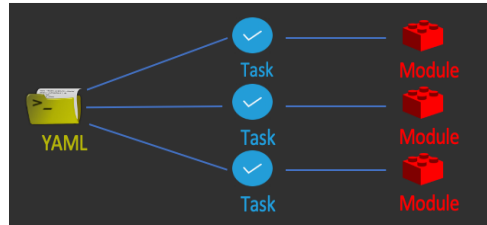


Policies



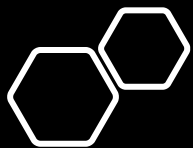
ANSIBLE

```
</> main.yml
1 - name: "apache installation"
2   yum:
3     name: "httpd"
4     state: "present"
5
6 - name: "apache service activation"
7   service:
8     name: "httpd"
9     state: "started"
10    enabled: yes
11
12 - name: "install php7 packages"
13   include: "php7-install.yml"
14   when: php_install|default(False)|bool
15
```



[YAML](#) (Yet Another Markup Language)

→ Each task invokes a module (python program). You can have multiple modules per task.



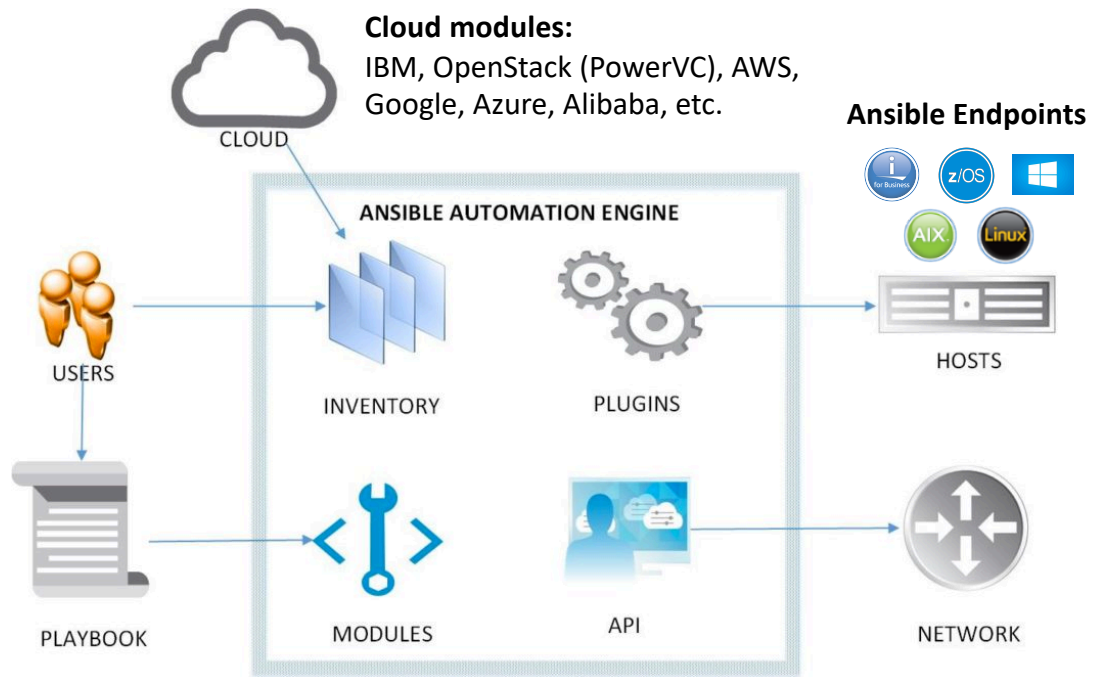
# What is Red Hat Ansible Engine?

Ansible Engine provides the core, agentless functionality of Ansible that everything else builds upon

Includes the basic building blocks of Ansible—the control node, managed nodes (endpoints), inventory, modules, tasks and playbooks

Commercial form of Ansible technology

Available for subscription purchase from Red Hat—from a POWER perspective, includes enterprise support options for AIX and IBM i managed endpoints (in June 2020)



**CERTIFIED INTEGRATION:**  
**Ansible and IBM Power Systems**

<https://www.ansible.com/integrations/infrastructure/ibm-power-systems>

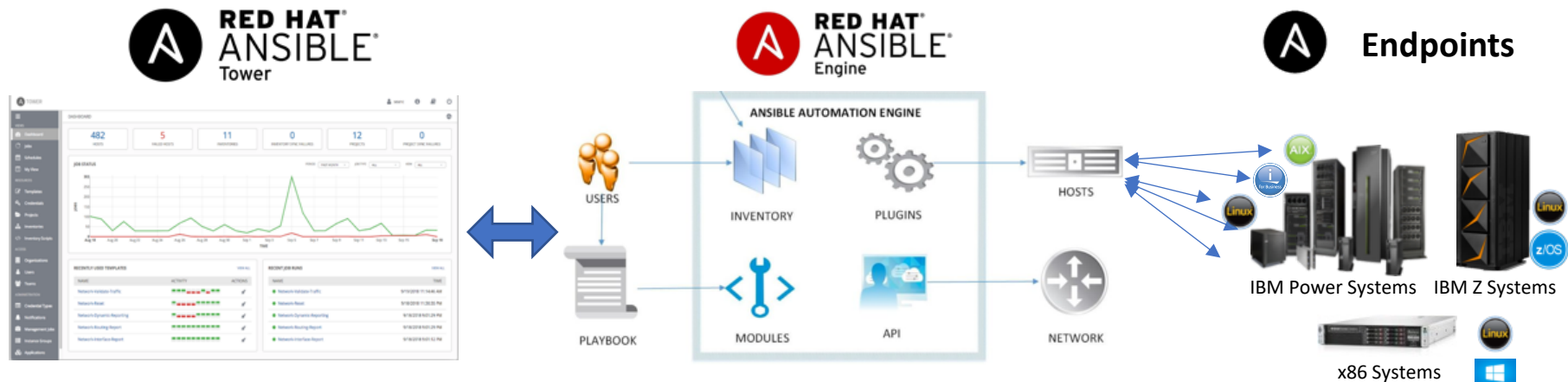


**RED HAT®**  
**ANSIBLE®**  
Engine



**Red Hat Ansible Engine supported  
on x86 Linux only — manages to endpoints**

# Ansible Automation for IBM Power Systems



1

## Red Hat Ansible Tower

- Enterprise-wide graphical control of Ansible estate



Supported on Linux

2

## Red Hat Ansible Engine

- Enterprise-wide control – i.e., runs playbooks



Supported on Linux

3

## Red Hat Ansible Endpoints

- Enterprise-wide automation; modules are executed here



Commercial support available from Red Hat



community support (at present)

# Ansible & IBM i

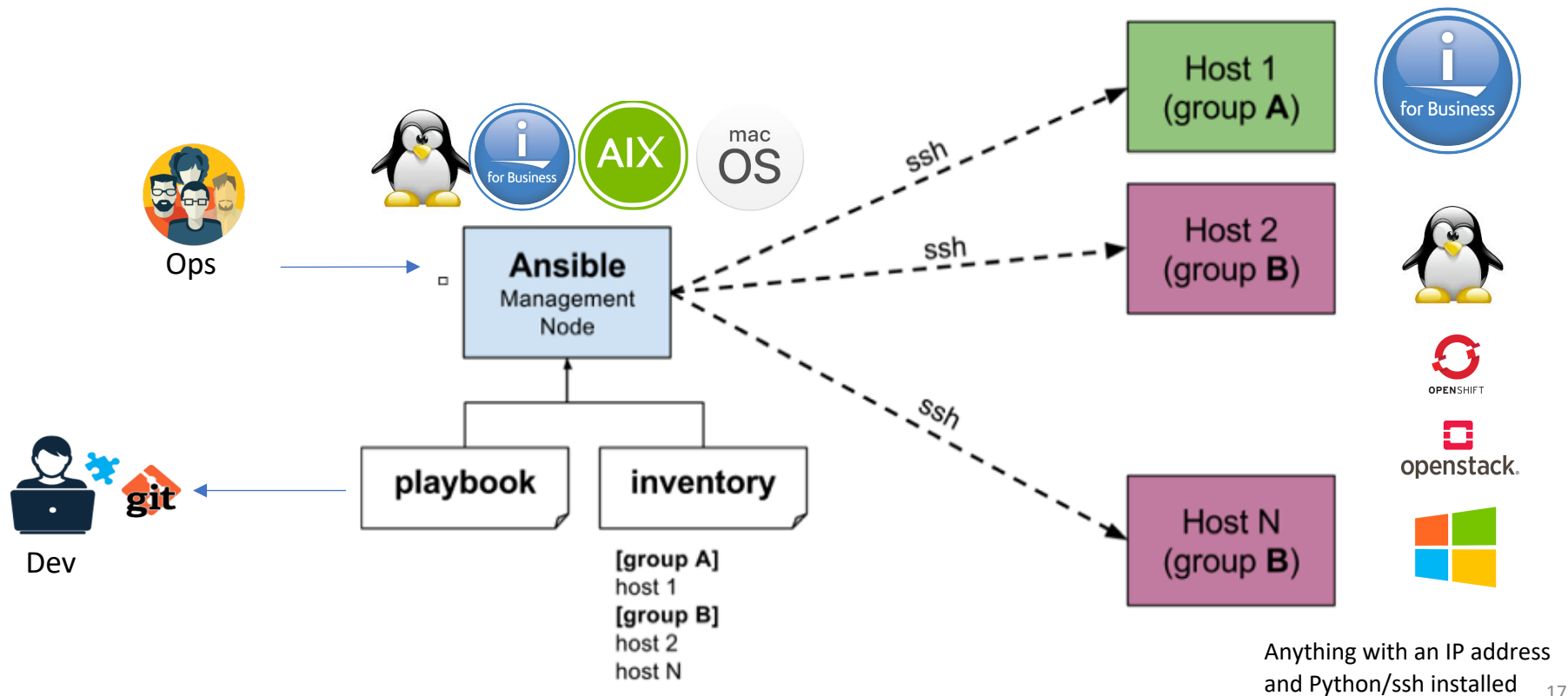
---



100 000 's Customers  
120 countries  
100,000's Systems  
70% Small and Mid-sized  
40 national languages  
98% of Fortune 100

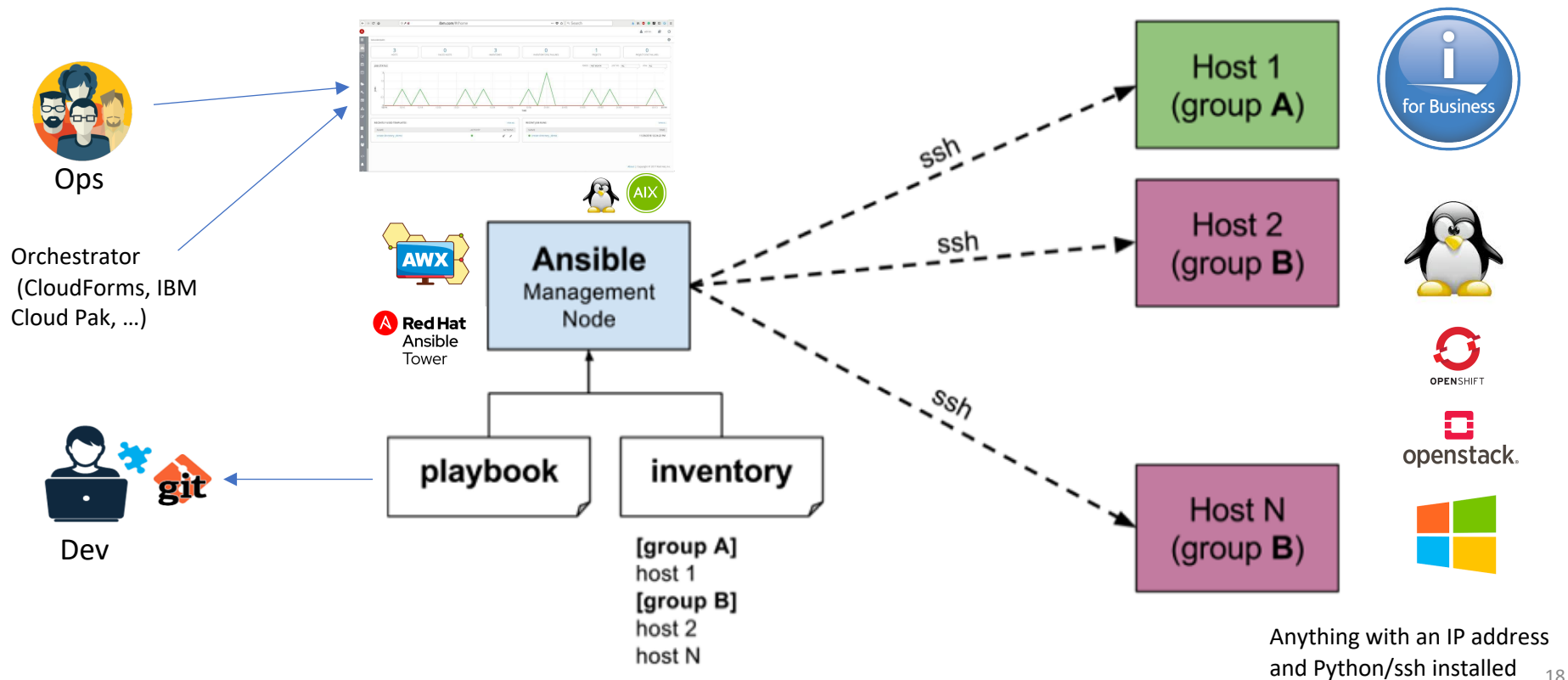
# Ansible Solution Architecture

## Basic Setup



# Ansible Solution Architecture

## Advanced Setup





# Ansible Tower



Ansible Tower = Supported Ansible solution for Enterprise

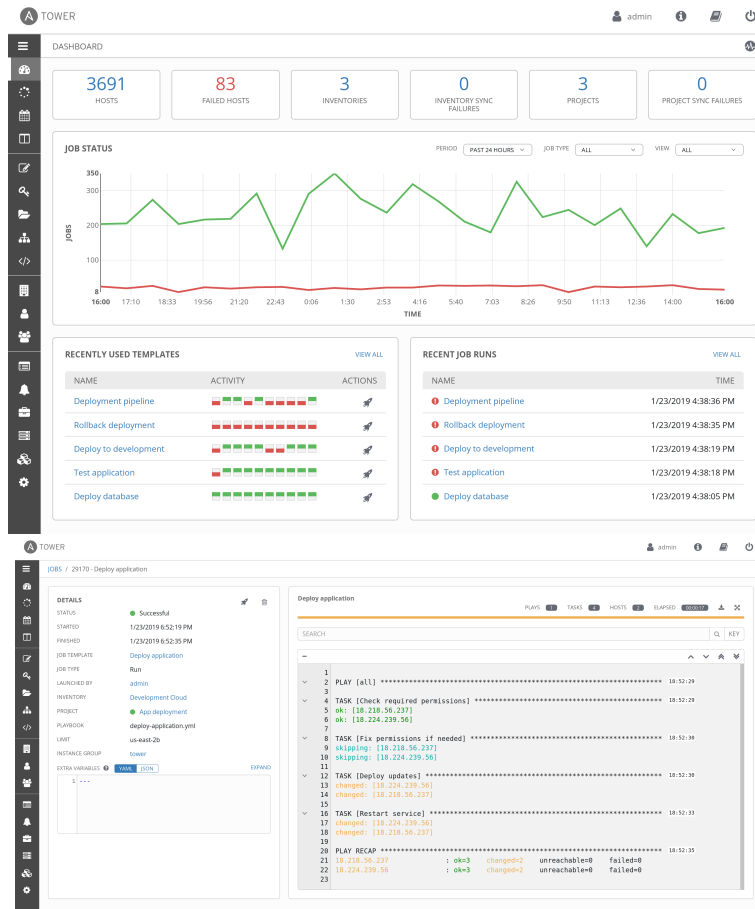
YOUR ANSIBLE DASHBOARD

REAL-TIME JOB STATUS UPDATES

MULTI-PLAYBOOK WORKFLOWS

WHO RAN WHAT JOB WHEN

SUPPORT IBM i TASKS



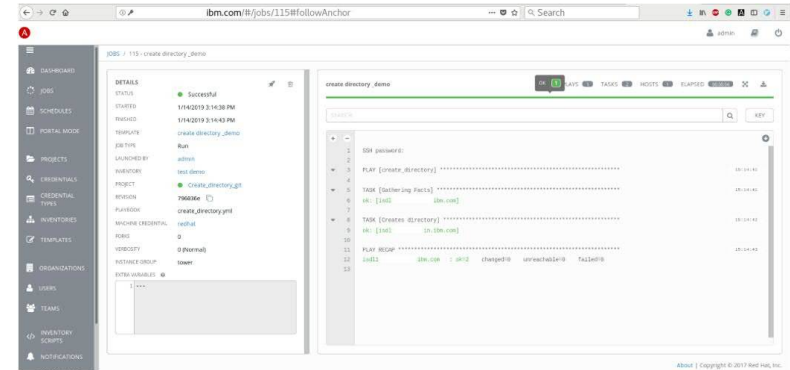
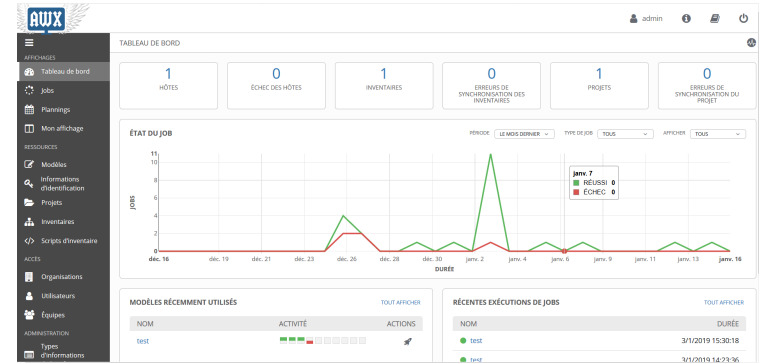
# Ansible AWX

<https://github.com/ansible/awx>



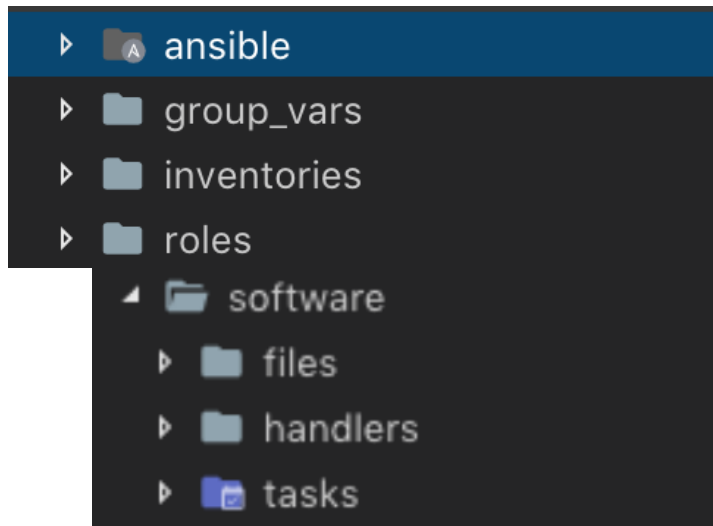
Ansible Tower upstream project. Community Support

YOUR ANSIBLE DASHBOARD  
REAL-TIME JOB STATUS UPDATES  
MULTI-PLAYBOOK WORKFLOWS  
WHO RAN WHAT JOB WHEN  
SUPPORT IBM i TASKS



<https://developer.ibm.com/technologies/systems/articles/automation-using-ansible-awx-gui/>  
<https://www.power-devops.com/post/installing-ansible-awx-on-ibm-aix>

# How to use Ansible - Intermediate



**group\_vars:** is the folder which will contain variable with it's value. Which you can use in your script. Variable can use username, password, software version, path or anything.

**inventories:** Inventories are definition for your cluster, VMs or Nodes. You can define you IP address or Domain name for all your VMs. In-case you want to SSH using public key or password based Authentication. Default : /etc/ansible/hosts

**roles:** This is the place where you define your automation.

Example:

IBMi > **files** (contains SAVF)

IBMi > **tasks** (contains your Playbooks i.e. your tasks in a script)

IBMi > **handlers** running operations on change: sometimes, you want a task to run only when a change is made on a machine  
Handlers are tasks that only run when notified.

**ansible.cfg :** system wide or user ansible config file  
Usually in /etc/ansible/ansible.config or ~/.ansible.cfg

# Ansible and IBM i : Modules



Write your first “playbook” in YAML format to describe what you want on your managed node inventory and Ansible will , for example :

- ✓ [Deploy or clone a new environment](#) on an IBM i VM on either a private or public cloud
- ✓ Install a new licensed program product or application version containing libraries, database and IFS artifacts
- ✓ Save or restore objects, manage servers or jobs and check and install PTFs
- ✓ Control your security settings, like managing user profiles and authorities, or check IFS rights. Ansible gathers facts and can remediate any security deviations.
- ✓ Orchestrate all of the above or a subset of these tasks

# Ansible and IBM i

## Core modules in PASE + IBM i Specific Modules



**Core Maintained** modules are maintained by the Ansible Engineering Team.

- Core modules are owned by RedHat and ship with Ansible installation.
- Many of these modules work for IBM i PASE environment.
- **Support PASE but not native IBM i.**

- command
- raw
- script
- shell
- pip
- yum
- pause
- wait\_for\_connection
- ~~•at~~
- authorized\_key
- gather\_facts
- ~~•group~~
- ~~•Mount~~

- ping
- ~~•reboot~~
- setup
- ~~•user~~
- assemble
- blockinfile
- copy
- fetch
- file
- find
- lineinfile
- stat
- synchronize
- git

# Ansible and IBM i

## Core modules in PASE + IBM i Specific Modules

- CL Commands
  - Executes CL commands and return general and detail job logs
- SQLs executions
  - Executes SQL statements and return the results
  - Queries – compare the returned single value result
  - Inserts / Updates / Deletes
  - Functions & Procedures
- Gathering facts and setup changes for IBM i
- Securities – authorization list, user profiles, grant object authorities
- Copy Objects, Fetch Objects, Find Objects
- Reply Message – query and reply
- Reboot system
- Network configurations
- Device configurations and management
- IASP configuration
- System Values, Environment variables, Etc.
- Submit / Schedule Jobs
- Manage fixes / PTFs / LPPs
- .... More to come!!! Check out

<https://github.com/IBM/ansible-for-i>

[ibmi at](#)

Schedule a batch job on a remote IBMi node.

[ibmi cl command](#)

Executes a CL command.

[ibmi copy](#)

Copy a save file from local to a remote IBMi node.

[ibmi display subsystem](#)

Display all currently active subsystems or currently active jobs in a subsystem.

[ibmi end subsystem](#)

End a subsystem.

[ibmi fetch](#)

Fetch objects or a library from a remote IBMi node and store on local.

[ibmi install product from savf](#)

Install the the licensed program(product) from a save file.

[ibmi lib restore](#)

Restore one library on a remote IBMi node.

[ibmi lib save](#)

Save one library on a remote IBMi node.

[ibmi object authority](#)

Grant, Revoke and Display the Object Authority.

[ibmi object restore](#)

Restore one or more objects on a remote IBMi node.

[ibmi object save](#)

Save one or more objects on a remote IBMi node.

[ibmi reboot](#)

Reboot IBMi machine.

[ibmi save product to savf](#)

Save the the licensed program(product) to a save file.

[ibmi script](#)

Execute a local cl/sql script file on a remote ibm i node.

[ibmi script execute](#)

Execute a cl/sql script file on a remote ibm i node.

[ibmi sql execute](#)

Executes a SQL non-DQL(Data Query Language) statement.

[ibmi sql query](#)

Executes a SQL DQL(Data Query Language) statement.

[ibmi start subsystem](#)

Start a subsystem.

[ibmi sync](#)

Synchronize a save file from current ibm i node A to another ibm i node B.

[ibmi synchronize](#)

Synchronize a save file from ibm i node A to another ibm i node B.

[ibmi uninstall product](#)

Delete the objects that make up the licensed program(product).

[ibmi user and group](#)

Create, Change and Display a user(or group) profile.

# Ansible and IBM i

## Playbooks Examples



- [enable-ansible-for-i](#)
  - `ibmi-install-rpm.yml`
  - `ibm-install-yum.yml`
  - `setup.yml`
- [ibmi-install-nodejs](#)
  - `ibmi-install-nodejs.yml`
- [ibmi-check-default-passwords.yml](#)
- [ibmi-cl-command-sample.yml](#)
- [ibmi-fix-group-check.yml](#)
- `ibmi-fix-repo-cum-package.yml`
- [ibmi-sysval-sample.yml](#)
- [query-iasp-sample.yml](#)
- `ibmi-sql-sample.yml`

<https://github.com/IBM/ansible-for-i>



# Galaxy – power\_ibmi

GALAXY

Home

Search

Community

AboutHelpDocumentationLogin

Community Authors> ibm> power\_ibmi

IBM

ibm

power\_ibmi

Ansible Content for IBM Power Systems - IBM I provides Ansible action plugins, modules, roles and sample playbooks to automate tasks on IBM i systems.

4.3 / 5 Score

3302 Downloads

Login to Follow

Issue Tracker

Repo

Docs Site

Details

Read Me

Content

Filter content...

Show: RolesModulesPlaybooksPlugins

apply\_all\_loaded\_ptfs

Role

Ansible role for applying all loaded ptfs

check\_ptf

Role

Ansible role for checking ptfs status according to given ptfs list

download\_individual\_ptfs

Role

Ansible role for downloading a list of individual ptfs using ibmi\_download\_fix module, and return st...

fix\_repo\_check\_ptf\_group

Role

Ansible role for getting the latest PTF group information, and check if the latest PTF group is alre...

fix\_repo\_extract\_ptf\_group\_info

Role

Ansible role for extracting and update ptf group's information into ptf\_group\_image\_info table in ca...

load\_apply\_ptfs

Role

Ansible role of load and apply a list of individual ptfs, and retrun status

sync\_apply\_individual\_ptfs

Role

Ansible role of tranfer a list of ptfs to an ibm i system, then load and apply. And return the statu...

apply\_ptf

Role

Ansible role for applying all loaded ptfs or a list of ptfs.

check\_ptfs\_by\_product

Role

Ansible role for checking product ptf

fix\_repo\_download\_add\_ptf\_group

Role

Ansible role for downloading a ptf group and then add download information into download\_status tabl...

load\_ptf

Role

Ansible role for loading a set of ptfs according to given ptfs list, and returned ptfs loaded status

Afficher un menu

# Ansible Support & Installation



Control node

- Ansible on Linux x86 : Community + Possible [Red Hat Subscription](#) and support
- Ansible on IBM i : Community + Possible [IBM TSS Support](#) (Open Source package)
- Ansible can be installed via your Linux distribution package manager
  - yum install ansible or apt install ansible
  - If not available, just install python-pip and dependencies and install it with “pip”
    - pip install ansible
- Clone the repository to your Ansible server
  - <https://github.com/IBM/ansible-for-i>
- Create your inventory file
  - example can be found in file examples/ibmi/host\_ibmi.ini



# Demo



<https://github.com/bmarolleau/ansible-for-i>

# Thank you

# Demonstration : Terraform with PowerVC / IBM Cloud, Self service with Cloud Automation Manager / Cloud Pak for MCM



**IBM Power Systems in the Multicloud Era : AIX/IBM i Terraform Automation and IBM Cloud Pak for MCM**

Part 2 : Automation with Terraform, IBM Cloud Pak for MCM (40 minutes)

<https://youtu.be/fHZiQiXSICQ>



<https://github.com/bmarolleau/devops-automation/tree/master/terraform/simple-vm-power-vs>  
<https://github.com/bmarolleau/pvs-terraform-lab>

## Assets & Videos:

- **IBM Power Systems in the Multicloud Era : Get Started with IBM Cloud Power Virtual Server**

Part 1 : Introduction demo to the IBM Cloud Power Virtual Server for AIX / IBM i (20 minutes)



[https://youtu.be/RywSfXT\\_LLs](https://youtu.be/RywSfXT_LLs)

- **Ansible & Power Systems : IBM i demo**

[MOP Demo – Ansible + IBM i](#)



# Thank you